



**Fermi National Accelerator Laboratory**

**TM-1643**

## **Mass Storage for Microprocessor Farms**

**H. Areti**

*Fermi National Accelerator Laboratory*

*P.O. Box 500*

*Batavia, Illinois 60510*

January 12, 1990



**Operated by Universities Research Association Inc. under contract with the United States Department of Energy**

## Mass Storage for Microprocessor Farms

H. Areti  
Fermi National Accelerator Laboratory  
P. O. Box 500  
Batavia, Il. 60510

### Abstract

Experiments in high energy physics require high density and high speed mass storage. Mass storage is needed for data logging during the online data acquisition, data retrieval and storage during the event reconstruction and data manipulation during the physics analysis. This paper examines the storage and speed requirements at the first two stages of the experiments and suggests a possible starting point to deal with the problem.

### Introduction

Over the past few years, the mass storage requirements of high energy physics experiments have been increasing rapidly. The experimental activity may be divided into three distinct phases: 1. data acquisition, 2. event reconstruction and 3. physics analysis. Traditionally, the mass storage medium at phase 1 was the 9 track magnetic tape; and phases 2 and 3 required both magnetic tape and disk storage. At the data taking stage of an experiment, the 9-track tape is giving way to 8 mm video tape cartridges for several reasons. The storage capacity of the 8 mm video cartridge is almost an order of magnitude higher than that of the 9-track tape, in a physical volume which is a fraction of that taken by the 9-track tape; the drives are smaller and less expensive. In addition, the low cost of the 8 mm video technology makes the implementation of multiple streams of data logging possible, allowing the experimenters to gather quantities of data which would have been unthinkable with 9-track tape technology. The video cartridge is a sequential storage device; however, at the data gathering and event reconstruction phases of an experiment this is not a detriment. The physics analysis stage, however, requires random access devices.

### Scope of the Problem

High level trigger decisions at large high energy physics experiments, such as those envisioned at the Superconducting Super Collider (SSC), will be made by online processor farms that are capable of delivering about a million mips of processing power (ref. 1 & 2). The event size at SSC experiments is expected to be 1 Mb and the data rate into the processor farm is assumed to be 10 Gbytes/s (ref. 2). At an event

logging rate of 100 Hz, the mass storage system at the online farm should be able to handle 100 Mbytes/s data rates; one thousand hours of running will accumulate 360 terabytes of data. Storing this data with today's 8 mm video tape cartridge (2 Gbytes/unit) will require 180,000 cartridges. The tape cartridges will cost approximately a million dollars and will take 1000 cubic feet of storage space. The 8 mm recording system with SCSI interface moves data at a rate of 200 Kbytes/s. The 100 Mbytes/s throughput requires nearly 500 drives and at 20 drives/ 19" rack, occupies 25 racks; i.e. 250 square feet of floor space. Offline event reconstruction at a million mips farm will require higher data rates from the mass storage system. For example, at the Collider Detector at Fermilab, a full event reconstruction (average event size of 150 Kbytes) requires 20 seconds on the VAX 11/780 (ref. 3). Assuming that an SSC event requires 1 second on a 1000 mips machine, a million mips farm processes nearly 1000 events per second. Thus, the data handling capacity of the mass storage system needs to be at least 1 Gbyte/s. The system performance at analysis stage will be limited more by the I/O bandwidth than the CPU power. One logistical problem that can not be overlooked is that the entire 500 drives need servicing at frequent (about half hour) intervals.

The numbers presented here are intended as order of magnitude estimates, and serve to identify two important problems with the mass storage at large experiments. One is an architectural problem of attaching the system to the processor farms, handling the data rates etc. and the other is a logistical problem of loading and unloading the storage media.

### Mass Storage System Architecture

A block diagram of the farm and the associated mass storage is shown in figure 1. The processor farm, discussed in a separate paper (ref. 1), consists of a large number of nodes. The mass storage controller mediates between the processors and the recording devices. The 'volumes' are mass storage devices: they may be sequential access devices or random access devices.

The mass storage controller (MSC) allows each node in the farm to access any one of the volumes. The data transfer protocol between the node and the MSC and the MSC and the mass storage device are unknown at the present. It is possible that the protocol between the node and the MSC is different from the protocol between the MSC and the mass storage. It is important that the protocols ( hopefully standard protocols) and the MSC as a system be able to handle 1 Gbyte/s data rate.

Obviously, if the intent is just to record or retrieve data, then a given node in the processor farm need not know which volume is being addressed. When the node wishes to write data to a volume, the data is sent to the MSC, which dispatches it to the first available volume. Similarly, when the node requires data, the MSC supplies it from the next idle device. Thus, every storage device is connected to every processor node of the farm. To achieve this type of connectivity at the desired data rates, the heart of the MSC needs to be a switch.

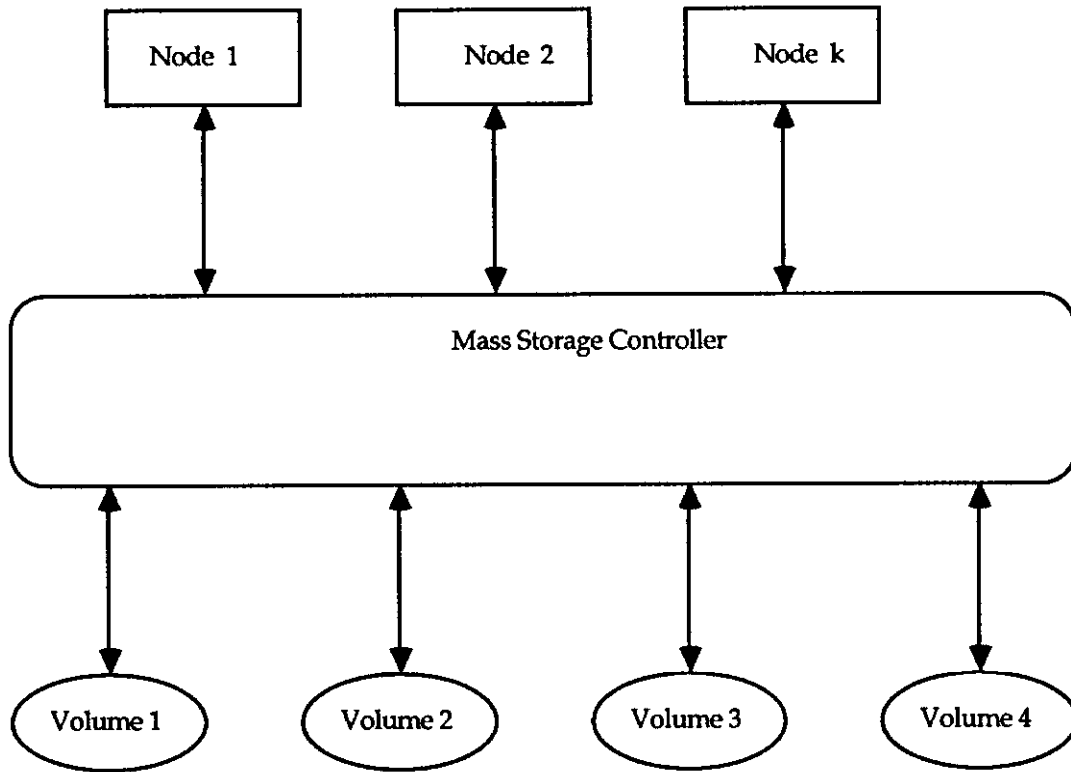


Figure 1.

The simple scheme of assigning the first available device to a requesting node is excessively limiting. Complex experiments tend to record data from several different triggers and it may be desirable to sort and store these events; in other words, a particular storage device stores a particular trigger type. In addition, the number of devices allocated to various triggers may change; the MSC should keep track of the assignments. Thus, the MSC is an intelligent switch, containing multiple data buffers and is capable of list processing (fig. 2). The FIFO buffers hold multiple events and aid in speed matching between the processor farm and the mass storage.

In this mass storage scheme, the nodes themselves are not aware of the status of the storage devices, since the nodes interact only with the MSC. Thus, the MSC has the additional task of monitoring the status of every storage device in the system. When a device needs servicing, (device full, malfunction, etc.), the MSC has to convey this information to the operator.

To/From Processor farm

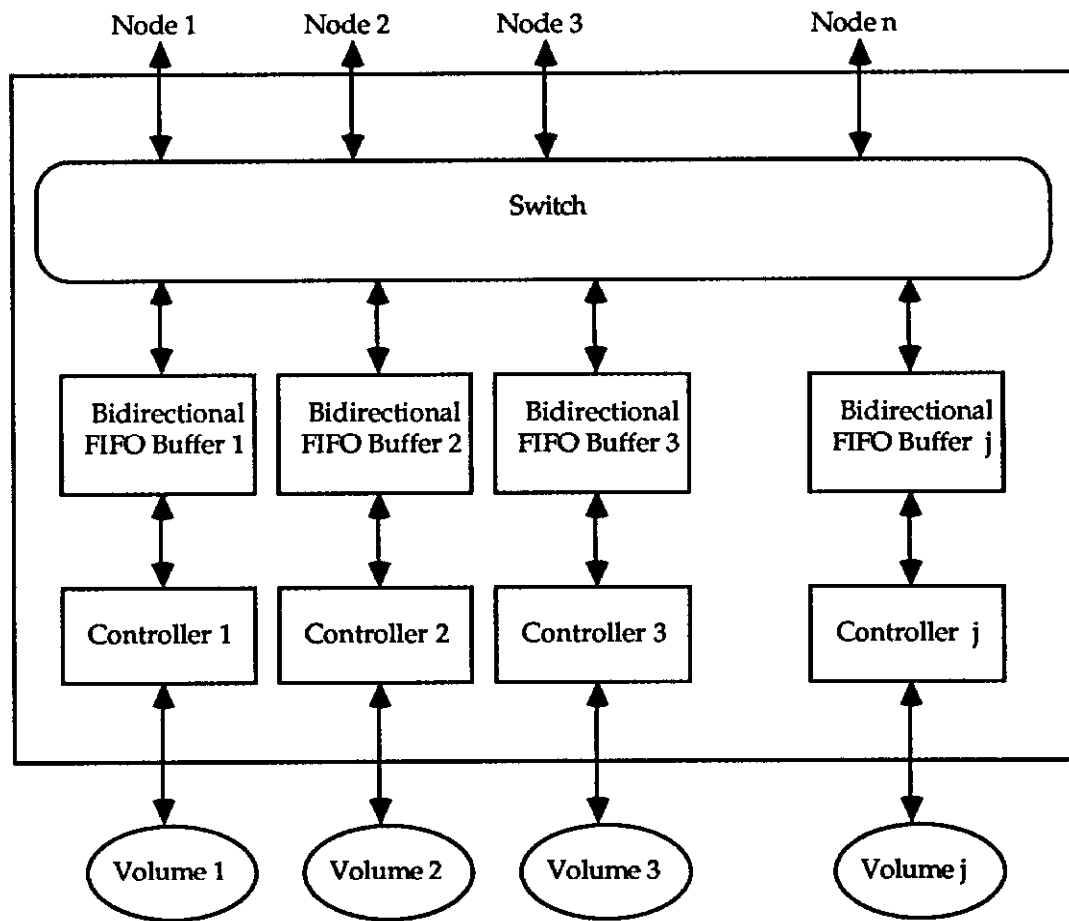


Figure 2.

Design of the MSC will be greatly simplified, if the controller/volume combination is a commercial entity. In this case, the FIFO buffers need to be intelligent enough to handle the controllers (requesting status, handling interrupts etc.).

This architecture could be extended to provide monitoring and graphics capabilities. One can imagine listening devices attached to the switch, which can store copies of events (fig. 3). These events can be processed and displayed by some other mechanism, without affecting the performance of the mass storage system.

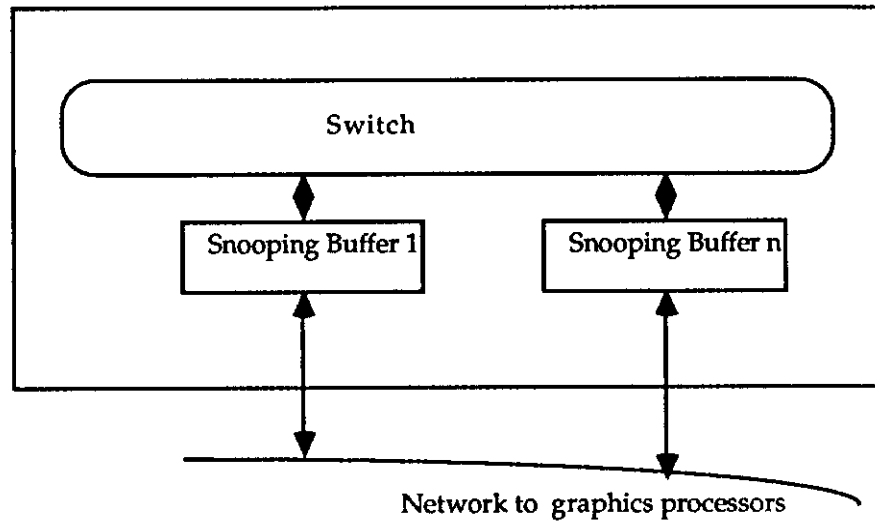


Figure 3.

### Conclusion

The data storage requirements at large experiments, such as those at the SSC, will require sophisticated mass storage technology. The very high performance microprocessor farms of the future will demand matching performance from the I/O devices. The most important logistical problem is the servicing of the mass storage devices. The frequency of servicing depends on the throughput of the system, the storage capacity of the devices in the system and the number of storage devices in the system. Automatic (robotic) or semi-automatic devices will be needed to solve this problem.

### Acknowledgments

The author wishes to thank his colleagues in the Advanced Computer Program for many valuable discussions over the years and Janice Enagonio, Richard Hance and Umesh Joshi for many helpful suggestions.

### References

1. H. Areti, Future Microprocessor Farms: Online and Offline.
2. Self-Routing Parallel Event Builder for SSC. Ed Barsotti, Private Communication.-
3. Umesh Joshi, Private Communication.